```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
import numpy as np
```

```
data = pd.read_csv("/content/zoo_data.csv")
```

```
data
```

	1	0	0.1	1.1	0.2	0.3	1.2	1.3	1.4	1.5	0.4	0.5	4	0.6	0.7	1.6	1.7
0	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	1	1
1	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	0	4
2	1	0	0	1	0	0	1	1	1	1	0	0	4	0	0	1	1
3	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1	1
4	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
95	1	0	0	1	0	0	0	1	1	1	0	0	2	1	0	1	1
96	1	0	1	0	1	0	0	0	0	1	1	0	6	0	0	0	6
97	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1	1
98	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	7
99	0	1	1	0	1	0	0	0	1	1	0	0	2	1	0	0	2

100 rows × 17 columns

```
y = data["1.7"]
y
```

```
0     1
1     4
2     1
3     1
4     1
     ..
95    1
96    6
97    1
98    7
99    2
Name: 1.7, Length: 100, dtype: int64
```

```
x = data.drop(["1.7"],axis=1)
```

```
x
```

	1	0	0.1	1.1	0.2	0.3	1.2	1.3	1.4	1.5	0.4	0.5	4	0.6	0.7	1.6
0	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	1
1	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	0
2	1	0	0	1	0	0	1	1	1	1	0	0	4	0	0	1
3	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1
4	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
95	1	0	0	1	0	0	0	1	1	1	0	0	2	1	0	1
96	1	0	1	0	1	0	0	0	0	1	1	0	6	0	0	0
97	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1
98	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
99	0	1	1	0	1	0	0	0	1	1	0	0	2	1	0	0

100 rows × 16 columns

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2)
```

```
# make the model of the classfier
classifier = DecisionTreeClassifier()
classifier.fit(x_train,y_train)
```

```
DecisionTreeClassifier()
```

```
# now we need to have y_pred
y_pred = classifier.predict(x_test)
y_pred
```

```
array([1, 6, 1, 4, 5, 1, 3, 2, 6, 1, 1, 4, 5, 7, 7, 6, 1, 2, 1, 2])
```

```
# now we need to check for the performance
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
```

```
print("accuracy is : ",accuracy_score(y_test,y_pred))
```

```
accuracy is :  0.95
```

```
print("Confusion Matrix:\n",confusion_matrix(y_test, y_pred))
```

```
Confusion Matrix:
 [[7 0 0 0 0 0 0]
 [0 3 0 0 0 0 0]
 [0 0 0 0 0 1 0]
 [0 0 0 2 0 0 0]
 [0 0 1 0 2 0 0]
 [0 0 0 0 0 2 0]
 [0 0 0 0 0 0 2]]
```

```
print("Classification Report:\n",classification_report(y_test, y_pred))
```

```
Classification Report:
                precision    recall  f1-score   support

           1       1.00      1.00      1.00         7
           2       1.00      1.00      1.00         3
           3       0.00      0.00      0.00         1
           4       1.00      1.00      1.00         2
           5       1.00      0.67      0.80         3
           6       0.67      1.00      0.80         2
           7       1.00      1.00      1.00         2

    accuracy                           0.90        20
   macro avg       0.81      0.81      0.80        20
weighted avg       0.92      0.90      0.90        20
```